

**Philosophische** Fakultät III

Sprach-, Literatur- und Kulturwissenschaften

Institut für Information und Medien, Sprache und Kultur (I:IMSK)  
Lehrstuhl für Informationswissenschaft

Modulprüfung Computational Intelligence

Modul: INF-BA-M09

SS 2021

Leitung: Stefan Kerscher/ Prof. Dr. Bernd Ludwig

**Trainieren eines neuronalen Netzes zur Prädiktion der Trajektorien von Fußgängern**

Pascal Strobel

Matrikelnummer: 2106133

6. Semester Medieninformatik/ Informationswissenschaft

E-Mail: pascal.strobel@stud.uni-regensburg.de

Abgegeben am 15.09.2021

Inhalt

[1 Einleitung 3](#_Toc82355264)

[2 Verwandte Arbeiten 4](#_Toc82355265)

[3 Methodik 5](#_Toc82355266)

[3.1 Problemstellung 5](#_Toc82355267)

[3.2 Datenvorverarbeitung 7](#_Toc82355268)

[3.2.1 Normalisieren der Daten 7](#_Toc82355269)

[3.2.2 Aufteilen der Daten in Trainings-, Validierungs- und Testdaten 8](#_Toc82355270)

[3.2.3 Standardisieren der Daten 8](#_Toc82355271)

[3.2.4 Formatieren der Daten 8](#_Toc82355272)

[3.3 Trainieren des neuronalen Netzes 10](#_Toc82355273)

[3.3.1 Rekurrente neuronale Netze (RNNs) 10](#_Toc82355274)

[3.3.2 Long short-term memory (LSTM) 10](#_Toc82355275)

[3.3.3 Netzwerkarchitektur des verwendeten Models 11](#_Toc82355276)

[4 Evaluation 11](#_Toc82355277)

[4.1 Metriken 11](#_Toc82355278)

[4.2 Resultate 12](#_Toc82355279)

[4.3 Fälle des Scheiterns des Ansatzes 12](#_Toc82355280)

[5 Diskussion 15](#_Toc82355281)

[6 Fazit 16](#_Toc82355282)

[Literaturverzeichnis 17](#_Toc82355283)

[Erklärung zur Urheberschaft 18](#_Toc82355284)

# Einleitung

Die Vorhersage der Trajektorien von Fußgängern gewinnt mit der ansteigenden Verbreitung von autonomem Fahren und sozialen Robotern, die den Menschen am Arbeitsplatz [Socially embedded learning of the], im Krankenhaus [HelpMate], im Museum [Experiences with an interactive museum], aber auch in den eigenen vier Wänden [Care-o-bot] unterstützen sollen, zunehmend an Bedeutung. Durch präzise Vorhersagen des Verhaltens von Fußgängern in der näheren Umgebung können mögliche Kollisionen frühzeitig erkannt und verhindert werden.

In dieser Arbeit wird mithilfe eines neuronalen Netzes (NN) ein datengesteuerter Lösungsansatz vorgestellt, um auf Basis vergangener Positionsdaten die zukünftige Trajektorie eines einzelnen Fußgängers vorherzusagen. Derartig datengesteuerte Ansätze für dieses Themengebiet haben sich vor allem in den letzten Jahren durch vielversprechende Resultate etabliert. Evlt Quellen

Die Arbeit gibt in Kapitel 2 zunächst einen Überblick über den aktuellen Forschungsstand zum Thema Trajektorien-Prädiktion von Fußgängern. In Kapitel 3 folgt die Erläuterung der zugrundeliegenden Problemstellung, sowie der vorgeschlagene Lösungsansatz. Dabei werden zuerst die angewandten Datenvorverarbeitungstechniken und anschließend die Architektur des neuronalen Netzes inklusive grundlegender, benötigter, mathematischer Konzepte dargelegt. Als nächstes erfolgt in Kapitel 4 neben der Evaluation der Performance auch eine qualitative Analyse für Szenarien, in denen der vorgestellte Ansatz scheitert. Die Diskussion, in der Limitierungen und mögliche Verbesserungsvorschläge aufgezeigt werden, und ein schlussendliches Fazit runden die Arbeit inhaltlich ab.

# Verwandte Arbeiten

Das „Social Force Model“ von [Social Force Model] bildet die Grundlage für physikbasierte Ansätze zur Prädiktion von Fußgänger-Trajektorien. Es beruht auf der Theorie, dass Fußgänger externen Kräften ausgesetzt sind und deren Verhalten durch diese mathematisch beschrieben und vorhergesagt werden kann. Ein weiterentwickeltes physikbasiertes Modell ist das von [BRVO] vorgestellte BRVO, welches eine bereits bestehende Fußgänger-Simulationsmethode (RVO) [RVO] mit Online-Lernen kombiniert, um so eine individualisierte Vorhersage für jeden Fußgänger zu treffen.

Durch das in den letzten Jahren starke Wachstum an Popularität von künstlichen neuronalen Netzen, aber auch durch die laut [Pedestrian Trajectory Prediction with CNN] gegebene Limitierung von physikbasierten Ansätzen, dass diese aufgrund handgefertigter Funktionen nur eine Teilmenge aller möglichen Verhaltensweisen von Fußgängern darstellen können, sind vor allem Deep Learning Ansätze, ganz besonders rekurrente neuronale Netze (RNN), stark in der Literatur vertreten.

RNNs bilden mit ihrem internen Gedächtnis eine sehr leistungsstarke Kategorie von künstlichen neuronalen Netzen und eignen sich besonders gut für die Arbeit mit sequenziellen Daten, beispielsweise bei der automatischen Spracherkennung [end-to-end], maschinellen Übersetzung [Neural machine translation] oder der Klassifikation von Bildern/ Videos [Look and Think twice].

Da die Aufgabe der Trajektorien-Prädiktion von Fußgängern als Zeitreihenproblem angesehen werden kann, ist auch in dieser Domäne die Verwendung von RNNs weit verbreitet, speziell die der „Long Short-Term Memory“ Zellen, vorgestellt von [LSTM]. Beispiele dafür sind die Arbeiten von [Social LSTM] [Pedestrian Trajectory Prediction in extremely crowded scenarios] [Dynamic and static context-aware LSTM] [A data driven model for Interaction]. Viele der Arbeiten zu diesem Thema beziehen neben den vergangen Positionsdaten eines Fußgängers, zusätzliche Kontextinformationen, wie beispielsweise die Trajektorien anderer Fußgänger [Social LSTM] [Pedestrian Trajectory Prediction in extremely crowded scenarios] [Dynamic and static context-aware LSTM] oder räumliche Informationen [Context-Aware Trajectory Prediction] [A data driven model for Interaction], mit in die Vorhersage ein. In dieser Arbeit ist das nicht der Fall, die Vorhersage der zukünftigen Bewegungsbahn erfolgt nur mithilfe bekannter, vergangener Positionen.

[An Empirical Evaluation of Generic] zeigen, dass faltende neuronale Netze kanonische rekurrente Architekturen in einer breiten Anzahl von Aufgaben übertreffen und folgern das nötige Überdenken der Assoziation von Sequenzmodellierung und RNNs. Derartige Architekturen sind derzeit bei der Vorhersage von menschlichen Trajektorien nur spärlich vorzufinden. Aussichtsreiche Resultate der Arbeiten von [Pedestrian Trajectory Prediction with Convolutional Neural Networks | Convolutional Neural Network for Trajectory Prediction] bestärken allerdings die obige Schlussfolgerung und drängen zu weiterer Forschung in dem Bereich.

# Methodik

In diesem Abschnitt wird zunächst die Problemstellung dargelegt und die informationswissenschaftliche Fragestellung abgeleitet. Anschließend wird ein möglicher Lösungsansatz präsentiert, welcher neben den verwendeten Vorverarbeitungstechniken der Daten und der Architektur des trainierten neuronalen Netzes auch die zugehörigen mathematischen Grundlagen erläutert.

## Problemstellung

Die Prädiktion der Trajektorien von Fußgängern entspricht der Vorhersage des zukünftigen Bewegungspfades basierend auf einer bestimmten Anzahl zuvor beobachteter Positionen. Die informationswissenschaftliche Fragestellung, die sich daraus ergibt, und für die diese Arbeit einen Lösungsansatz liefert, lässt sich demnach folgendermaßen formulieren:

„Wie kann anhand bekannter, vorheriger Positionen die zukünftige Trajektorie eines Fußgängers bestimmt werden?“

Zur Beantwortung der Frage soll auf Basis eines gegebenen Datensatzes, welcher Daten von sich bewegenden Fußgängern enthält, ein neuronales Netz trainiert und eine Regression durchgeführt werden. Jeder Datenpunkt im Datensatz besteht dabei aus einer Zeitreihe mit insgesamt 20 Zeitschritten und den dazugehörigen Werten für die aktuelle x- und y-Koordinate (in Metern) des Fußgängers. Die zeitlichen Intervalle zwischen zwei Zeitpunkten innerhalb eines Datenpunktes betragen durchgehend 400ms. Neben den Koordinaten sind keine weiteren kontextuellen Informationen über den Fußgänger oder seine Umgebung bekannt.

Das trainierte NN soll anschließend dazu genutzt werden, mithilfe der Positionsdaten aus den ersten acht Zeitpunkten eines Datenpunktes, die zukünftigen zwölf Positionen des Fußgängers vorherzusagen.

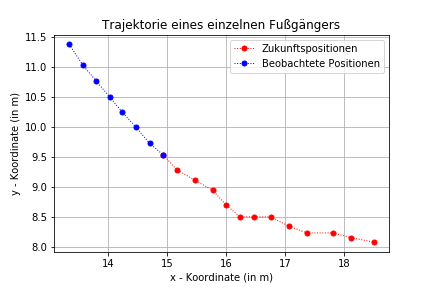


Abbildung 1: Beispiel-Trajektorie eines Fußgängers. Mithilfe der ersten acht beobachteten Positionen (blau) sollen die zukünftigen zwölf (rot) vorhergesagt werden. (eigene Darstellung)

Alle Grafiken mit [Matplotlib] erstellt 🡪 Fussnote machen

Diese acht-zu-zwölf-Verteilung ist gängig in der Literatur und beispielsweise in den Arbeiten von [Social LSTM: Human Trajectory Prediction in Crowded Spaces | Pedestrian Trajectory Prediction with Convolutional Neural Networks | Convolutional Neural Network for Trajectory Prediction] zu finden. Die zugrundeliegende Aufgabe wird in Abb. 1 zusätzlich visualisiert.

## Datenvorverarbeitung

Damit das neuronale Netz trainiert und ausgewertet werden konnte, mussten zunächst die gegebenen Daten entsprechend vorverarbeitet werden. Im Folgenden werden alle angewandten Schritte und Techniken dargelegt.

### Normalisieren der Daten

Der ursprüngliche Datensatz enthält die Positionen der Fußgänger als absolute Koordinaten. Da allerdings kein Ursprung für diese Koordinaten festgelegt ist, können die Bewegungsdaten unterschiedlicher Fußgänger absolut weit auseinander liegen (Fußgänger 1 z. B. im Bereich 1 bis 10m, Fußgänger 2 im Bereich 90 bis 100m).

[Pedestrian Trajectory Prediction with Convolutional Neural Networks] identifizieren neben dem naiven Ansatz, die absoluten Koordinaten ohne weitere Normalisierung zu verwenden, drei mögliche Normalisierungstechniken:

* Koordinaten haben den Ursprung im ersten beobachteten Zeitpunkt
* Koordinaten haben den Ursprung im letzten beobachteten Zeitpunkt
* Relative Koordinaten

Alle drei Ansätze führen dabei zu einer Performancesteigerung des in der Arbeit vorgestellten, faltenden neuronalen Netzes für unterschiedliche Datensätze, was die Notwendigkeit der Normalisierung der Daten betont.

In dieser Arbeit werden die Daten gemäß dem zuletzt aufgeführten Stichpunkt normalisiert. Das bedeutet, dass das neuronale Netz nur mit den relativen Änderungen der x- und y-Koordinaten zwischen zwei Zeitpunkten arbeitet, welche im weiteren Verlauf der Arbeit als Delta-x (dx) und Delta-y (dy) bezeichnet werden. Zur Bestimmung der Delta Werte wurde für jeden Zeitpunkt eines Datenpunktes und für alle Datenpunkte im Datensatz die Differenz der x- und y-Koordinaten vom aktuellen Zeitpunkt und zum vorherigen Zeitpunkt berechnet.

Das führt zu einer nötigen Spezialisierung der Anforderungen an das neuronale Netz: Aus den ersten acht Zeitschritten eines Datenpunktes werden je sieben dx- und dy-Werte berechnet. Das neuronale Netz soll demnach Zeitreihen der Länge sieben und mit den zwei Features dx und dy entgegennehmen und dafür die nächsten zwölf dx- und dy-Werte vorhersagen. Fussnote: Die Umwandlung in die eigentlichen Koordinaten erfolgen ganz einfach durch das addieren der jeweiligen vorhergesagten Delta-Werte auf die vorherigen Koordinaten

### Aufteilen der Daten in Trainings-, Validierungs- und Testdaten

Die normalisierten Datenpunkte wurden nach dem Zufallsprinzip in Trainings- (80%), Validierungs- und Testdaten (je 10%) aufgeteilt. Zur Vermeidung von Data-Leakage erfolgte diese Aufsplittung vor der Standardisierung der Daten.

### Standardisieren der Daten

Um sicherzustellen, dass die Wertebereiche der zwei Merkmale dx und dy nicht zu unterschiedlich sind, werden diese mithilfe der z-Transformation standardisiert, sodass für jedes Merkmal ein Erwartungswert von 0 und eine Standardabweichung von 1 vorliegt. Der -Score des Samples lässt sich mit folgender der Formel berechnen:

ist dabei das arithmetische Mittel und die Standardabweichung aller Trainingssamples.

Eine Verlagerung der Mittelwerte der Inputvariablen gegen null, wie es bei der z-Transformation der Fall ist, wird auch von [EfficientBackprop] empfohlen, da jede Verschiebung des durchschnittlichen Inputs weg von null die Updates der Gewichte in eine bestimmte Richtung verzerrt und damit den Lernvorgang des neuronalen Netzes verlangsamt.

StandardScaler nur auf Trainingsdaten gefittet und dann derselbe für Validierungs- und Testdaten angewandt.

Die Standardisierung der Trainingsdaten, sowie die später folgende Umkehrung der z-Scores erfolgte mithilfe der StandardScaler- Klasse der Scikit-learn Bibliothek. (evtl. als Fußnote)

### Formatieren der Daten

Zur Bestimmung der nächsten zwölf Positionen eines Fußgängers wird in dieser Arbeit die Strategie der sequenziellen Vorhersage angewandt. Das bedeutet, das NN prädiziert basierend auf sieben dx und dy Werten immer nur den nächsten dx und dy Wert, statt alle zwölf auf einmal. Die neugewonnenen Informationen nach einer Vorhersage werden anschließend an die vorherigen sechs Input-Positionen angehängt (der erste Wert fällt weg, da das NN als Input eine Zeitreihe der Länge sieben erwartet), was wiederum den Input für die nächste Prädiktion bildet. Dieser Vorgang wird zwölfmal wiederholt.

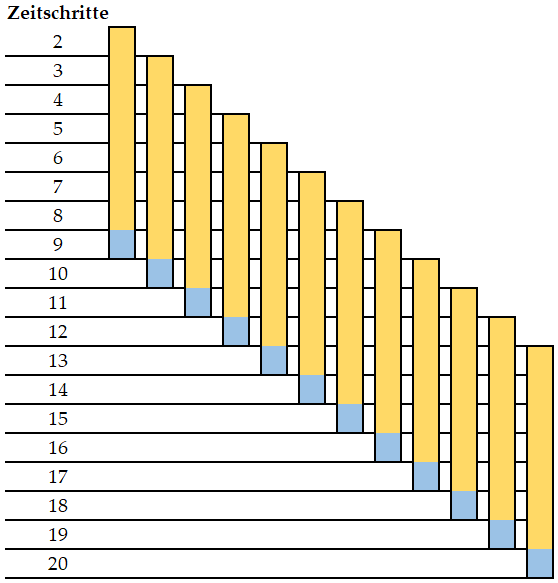


Abbildung 2: Prinzip der Zerlegung eines Datenpunktes mit 20 Zeitschritten in zwölf Trainingssamples (dargestellt durch farbige Balken). Jedes Trainingssample besteht aus einer Inputsequenz (orange) und aus dem Orakel (blau). Der erste Zeitschritt ist irrelevant, da für diesen keine Delta Werte berechnet werden können. (eigene Darstellung)

Damit das NN für diese Aufgabe trainiert werden konnte, mussten die Trainings- und Validierungsdaten, entsprechend angepasst werden: Jeder Datenpunkt des Trainingsdatensatzes wurde in zwölf Trainingssamples zerlegt. Die Delta Werte von sieben aufeinanderfolgende Zeitschritten bilden dabei immer den Input, und der darauffolgende dx und dy Wert den gewünschten Output/ das Orakel zum Input. Abb. 2 verdeutlicht das Prinzip der Zerlegung.

## Trainieren des neuronalen Netzes

In diesem Kapitel werden zunächst Hintergründe für grundlegende Designentscheidungen, und anschließend die eigentlich verwendete Netzwerkarchitektur, erläutert.

### Rekurrente neuronale Netze (RNNs)

Rekurrente neuronale Netze eignen sich sehr gut für das Arbeiten mit sequenziellen Daten und liefern in unterschiedlichen Domänen aussichtsreiche Resultate [evl. Quellen]. Im Bereich der Prädiktion von Fußgänger-Trajektorien haben sie sich sogar als state-of-the-art Ansatz etabliert [Quellen].

Wie auch klassische Feedforward- oder faltende neuronale Netze, werden Trainingsdaten zum Lernen benötigt. Der Unterschied liegt im „Gedächtnis“ von RNNs, welches ermöglicht, dass Informationen aus früheren Inputs verwendet werden können, um die aktuellen Inputs und Outputs zu beeinflussen. Zudem hängt die Ausgabe von RNNs von den vorherigen Elementen innerhalb der Sequenz ab, während konventionelle tiefe neuronale Netze die Unabhängigkeit von Input und Output annehmen. Eine weitere spezielle Eigenschaft ist, dass sie sich innerhalb jeder Schicht des Netzwerkes den gleichen Gewichtungsparameter teilen, und nicht über jeden Knoten hinweg unterschiedliche Gewichtungen aufweisen.

RNNs nutzen den Backpropagation-Through-Time (BPTT) Algorithmus, welcher sich vom herkömmlichen Backpropagation Algorithmus in der Hinsicht unterscheidet, dass er die Fehler bei jedem Zeitschritt summiert, was bei konventionellen Feedforward-Netzwerken nicht der Fall ist, da hier die Gewichtungsparameter nicht über die Schichten geteilt werden.

Das führt dazu, dass RNNs vom Problem des verschwindenden Gradienten betroffen sein können. Das Problem tritt auf, wenn der Gradient, welcher der Steigung der Verlustfunktion entlang der Fehlerkurve entspricht, zu klein ist. Ist dies der Fall, so wird er kontinuierlich kleiner, die Gewichtungsparameter werden so lange aktualisiert, bis sie verschwindend gering sind, und das führt wiederum zur Stagnation des Lernvorgangs des Netzes.

### Long short-term memory (LSTM)

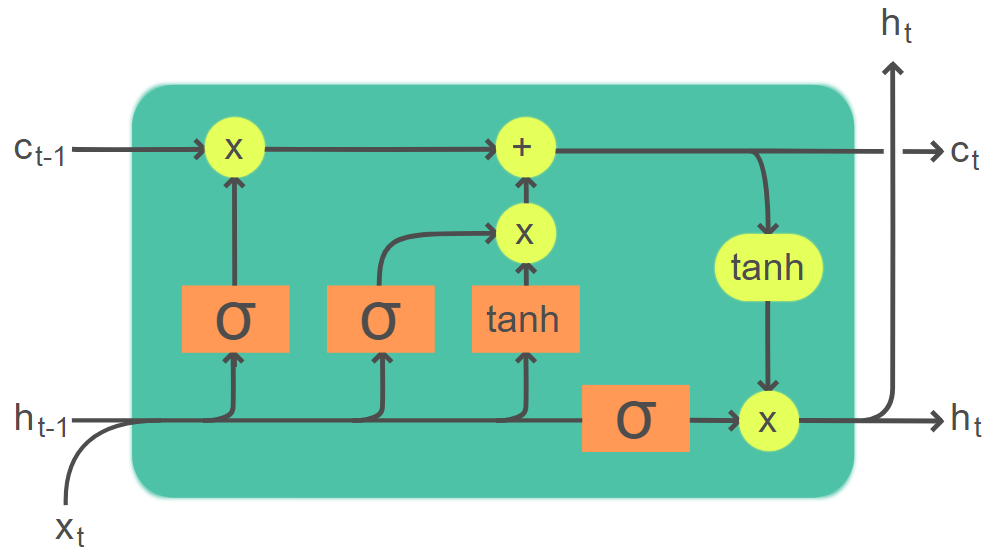


Abbildung : Aufbau einer einzelnen LSTM Zelle (https://en.wikipedia.org/wiki/Long\_short-term\_memory)

Das von [LSTM] eingeführte LSTM ist eine beliebte RNN-Architektur, welche das Problem des verschwindenden Gradienten behandelt.

<https://www.youtube.com/watch?v=QciIcRxJvsM&t=798s>

### Netzwerkarchitektur

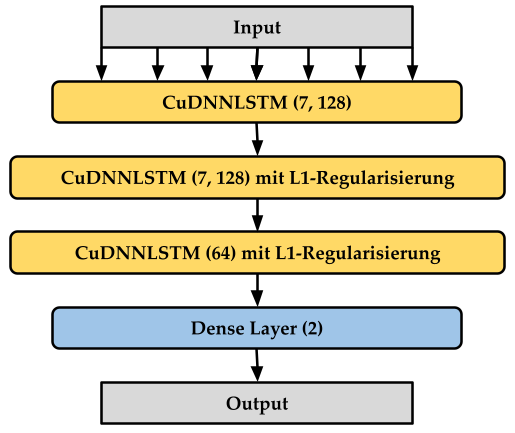


Abbildung : Generelle Architektur des Models. Klammern enthalten die Form des Outputs. (eigene Darstellung)

Mithilfe der Keras Bibliothek wurde ein sequenzielles Model, welches aus insgesamt vier Schichten aufgebaut ist, erstellt.

Eine lineare Aktivierungsfunktion der Dense Layer wurde gewählt, da sowohl positive, als auch negative Werte vom neuronalen Netz als Ergebnisse benötigt werden.

* Was ist LSTM/ CuDNNLSTM 🡪 Fussnote, dass CUDNN verwendet wurde:

CuDNNLSTM: optimierte LSTM Implementierung der NVIDIA CUDA Deep Neural Network (CuDNN) Bibliothek. 🡪 Fußnote

* Was ist eine Dense Layer
* Was sind units
* Welche methoden wurden gegen Overfitting angewendet? 🡪 L1-Regularisierung 🡪 wie funktioniert das?
* Learning rate
* Epochen
* Adam optimizer
* Loss funktion

# Evaluation

In diesem Kapitel erfolgt die Auswertung der Performance des vorgestellten Netzwerks anhand von zwei gängigen Metriken. Zudem werden mehrere Fehlerfälle identifiziert, welche aus einer qualitativen Analyse der prädizierten Trajektorien hervorgehen.

## Metriken

Ebenso wie in verwandter Literatur von bspw. [Pedestrian Trajectory Prediction with Convolutional Neural Networks | Convolutional Neural Network for Trajectory Prediction] werden zur Auswertung der Performance die Metriken „Average Displacement Error“ (ADE), erstmalig eingeführt von [You’ll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking], und „Final Displacement Error“ (FDE), eingesetzt.

Der ADE eines Fußgängers ist die durchschnittliche euklidische Distanz zwischen den prädiktierten Positionen und der Ground-Truth für alle vorhergesagten Zeitschritte eines Fußgängers. Der ADE lässt sich mit folgender Formel berechnen:

ist der Zeitschritt der letzten Beobachtung des Fußgängers (in unserem Fall gilt = 8), der final zu vorhersagende Zeitschritt eines Datenpunktes (in unserem Fall gilt = 20). ist die vorhergesagte und die tatsächliche Position zum Zeitpunkt t. repräsentiert die euklidische Distanz.

Der FDE eines Fußgängers ist die euklidische Distanz zwischen der prädizierten Position und der zugehörigen Ground-Truth zum finalen Zeitpunkt und lässt sich folgendermaßen berechnen:

## Resultate

Die Performance des trainierten RNN wurde anhand des zu Beginn abgespaltenen Testdatensatzes (10% der Gesamtdaten), welcher die Trajektorien von insgesamt 540 Fußgängern umfasst, ausgewertet, indem für jeden Datenpunkt sowohl ADE als auch FDE bestimmt wurde. Damit eine gesammelte Aussage für die Performance des Systems gemacht werden kann, wurde der Durchschnitt der jeweiligen Metrik von alle 540 Datenpunkte berechnet, was zu folgendem Ergebnis führt:

Das RNN erzielt für die Testdaten einen durchschnittlichen ADE von 0,763 und einen durchschnittlichen FDE von 1,561.

## Fälle des Scheiterns des Ansatzes

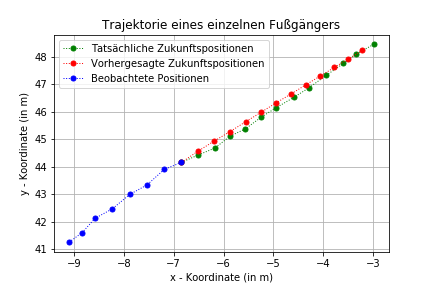


Abbildung 5: Präzise Vorhersagen durch triviales Fußgängerverhalten (eigene Darstellung)

Nicht für alle Fußgänger ist die Vorhersage der zukünftigen Trajektorie so trivial wie in Abb. 1. Abrupte Richtungswechsel oder Geschwindigkeitsänderungen sorgen für ein oft unberechenbares Verhalten von Fußgängern und damit zu Einbußen bei der Performance. Unzuverlässige Prädiktionen des vorgestellten Systems sind vor allem bei den folgenden drei Szenarien zu beobachten.

1. Fußgänger ändert plötzlich Richtung oder Geschwindigkeit, ohne einen Hinweis darauf:

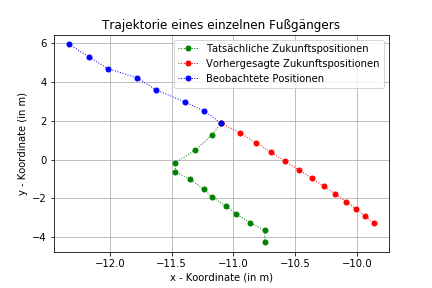


Abbildung 6: Überraschende Richtungsänderung (eigene Darstellung)

Die beobachtete Trajektorie des Fußgängers in Abb. 2 gibt keinerlei Hinweise auf eine mögliche Richtungsänderung. Die bestmögliche Vorhersage, die vom System (aber auch vom Menschen) getroffen werden kann, ist das Weiterführen dieser Bewegungsbahn. Damit derartige Bewegungsmuster besser prädiziert werden können, sind zusätzliche Kontext-Informationen (bspw. Straßenverlauf, Aufeinandertreffen mit anderen Fußgängern, o. ä.) notwendig.

1. Fußgänger ändert mehrfach seine Richtung oder Geschwindigkeit:

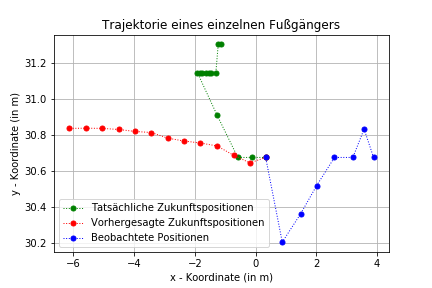


Abbildung 7: Mehrfache Richtungs- und Geschwindigkeitsänderungen (eigene Darstellung)

Nicht nur plötzliche, unerwartete, sondern auch häufige Richtungs- und Geschwindigkeitsänderungen in den beobachteten Positionen, führen zu Schwierigkeiten bei der Vorhersage (siehe Abb. 3). Ein richtiges Muster ist dabei auch für den Menschen schwer erkennbar. Wie auch schon im ersten Fall, sind zusätzliche Kontext-Informationen nötig, die eine zuverlässigere Prädiktion erlauben.

1. Fußgänger ändert seine Position nicht:

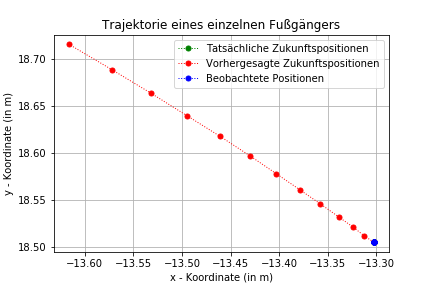


Abbildung 8: Keine Positionsänderung (eigene Darstellung)

Ebenfalls schlechte Vorhersagen des Systems sind zu beobachten, wenn sich der Fußgänger nicht bewegt (siehe Abb. 4). In derartigen Szenarien scheint das System nicht gelernt zu haben, keinerlei Bewegung in x- und y-Richtung vorherzusagen. Stattdessen werden Werte für dx und dy prädiziert, deren Betrag geringfügig von Null abweicht. Durch das sequenzielle Vorgehen bei der Bestimmung der nächsten zwölf dx- und dy-Werte akkumuliert sich dann der Fehler und führt zu zunehmenden Abweichungen zwischen Vorhersage und Ground-Truth.

# Diskussion

Das vorgestellte System liefert zufriedenstellende Ergebnisse für triviale Bewegungsbahnen. Allerdings deuten sich einige Schwierigkeiten bei der Vorhersage von komplexerem Fußgängerverhalten, was nicht untypisch für Fußgänger ist, an.

Eine Möglichkeit zur Optimierung des Systems ist die automatische Feinabstimmung von Hyperparametern, da dies bisher manuelle durchgeführt wurde, und somit nur eine kleine Anzahl an Hyperparameterwerten und Kombinationen getestet werden konnte.

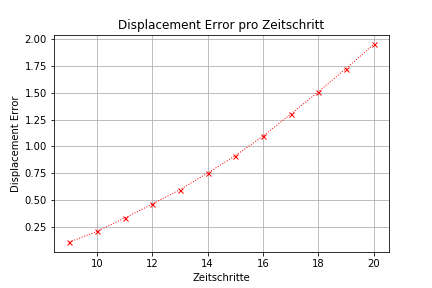


Abbildung 9: Akkumulieren des Fehlers für ferner in der Zukunft liegende Vorhersagen (eigene Darstellung) 🡪 bessere Beschriftungen!

Des Weiteren ist eine Verbesserung der Performance zu erwarten, wenn die zukünftigen Positionen nicht sequenziell, sondern per Multi-Output-Prädiktion auf einen Schlag vom neuronalen Netz bestimmt werden. Bei sequenziellen Vorhersagen akkumuliert sich der Fehler (schön dargestellt in Abb. 6), d. h. je weiter die Vorhersage in der Zukunft liegt, desto größer der Fehler (siehe Abb. 7), da diese auf möglichen vorherigen Fehlern basiert. Ein Multi-Output Netzwerk ist diesbezüglich resistenter.

Auch eine Erweiterung der verfügbaren Trainingsdaten durch das Sammeln weiterer Daten, oder durch „Data Augmentation“ (bspw. durch das Rotieren/ Spiegeln/ Anwenden von Gaußschen Rauschen auf Fußgänger-Trajektorien [Pedestrian trajectory prediction]), kann sich positiv auf die Performance auswirken. Eine größere Anzahl an Trainingsbeispielen für bestimmte Fußgängerverhaltensweisen führt dazu, dass das neuronale Netz diese besser lernen kann.

Weiterhin ist anzumerken, dass das vorgestellte System nur für einen Datensatz trainiert und getestet wurde. Damit ein Vergleich zu bereits bestehenden Ansätzen aus verwandten Arbeiten gezogen werden kann, muss die Performance auch für andere, in der Literatur etablierte Datensätze (bspw. ETH [Depth and appearance], UCY [Crowds by example] oder TrajNet [An evaluation of trajectory prediction and notes on trajNet]) bewertet werden.

# Fazit

Diese Arbeit präsentiert einen möglichen Lösungsansatz zur sequenziellen Prädiktion von Fußgänger-Trajektorien mithilfe eines rekurrenten neuronalen Netzes auf Basis von LSTM. Dabei wurden alle angewandten Datenvorverarbeitungsschritte, die Architektur des trainierten Netzes, sowie benötigte mathematische Grundlagen erläutert. Das System wurde anschließend bezüglich seiner Performance evaluiert und erzielt einen ADE von 0,763 und einen FDE von 1,561. Eine qualitative Analyse offenbart aussichtsreiche Ergebnisse für triviale Bewegungsbahnen, nicht-triviale führen allerdings zu unzuverlässigen Vorhersagen, was die Notwendigkeit der Optimierung des Systems betont. Die Arbeit zeigt zudem Limitierungen, sowie mögliche Lösungsansätze zur Optimierung des Systems, auf.

Literaturverzeichnis

Erklärung zur Urheberschaft

Wir haben die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie alle Zitate und Übernahmen von fremden Aussagen kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Die vorgelegten Druckexemplare und die vorgelegte digitale Version sind identisch.



Regensburg, 14.09.2021

|  |  |  |
| --- | --- | --- |
| Ort, Datum |  | Unterschrift |